SETI-CPS-ES Option
Critical Real-Time Systems - Exam

Florian Brandner - Laurent Pautet

2023-2024

2h00 – **Handwritten documents allowed**
Electronic communication devices (phones, ...) are not allowed
You can reply in English or French
The clarity and style of your explanations will be considered during grading
Only readable, precise, and sufficiently explained answers will be considered
Points for grading may vary

The exam is composed of independent parts that can be treated in any order. You are asked to write each part on a separate copy.

Part 1 : real-time scheduling

1 Code Coverage (3 points)

Question 1. (1 point) Code coverage [course]
Indicate what should be done by a test suite to guarantee the code coverage according to the approaches 1) Statement Coverage, 2) Decision Coverage and 3) Modified Condition / Decision Coverage. Particular attention will be paid to the clarity of your explanations.

course

Question 2. (2 points) Applying approaches SC, DC, MC/DC
Consider the code below. Illustrate each of these approaches by providing possible values of a, b, and c for the conditions. Particular attention will be paid to readability and clarity.

```
if ((a == 0) && ((b != 1) || (c == 2))) {F1(a, b, c)}; F2(a, b, c) ;
```

A	B	C	D
T	T	F	T
F	T	F	F
T	F	F	F
T	F	T	T

2 Partitioned scheduling on a dual-core processor (4 points)

We define a partitioned architecture with an identical two-core processor and five deadline implicit tasks whose periods and budgets are given in the table below. The scheduler is a fixed-priority partitioned scheduler.

Task	Period	Core	Budget
DISPLAY_1	6ms	A	4ms
SENSOR_3	20ms	A	4ms
DISPLAY_2	3ms	B	2ms
SENSOR_1	5ms	B	2ms
SENSOR_2	5ms	B	2ms

Question 1. (1 point) Demonstrate that, in the proposed configuration (assignment of tasks to processors), at least one task does not meet its deadline.

> The utilization rate on B is $\frac{22}{15} > 1$.

Question 2. (1 point) Demonstrate that no configuration can schedule this set of tasks.

> The utilization rate of the set is $\frac{7}{3} > 2$.

Question 3. (1 point) We are going to accelerate processor A and maintain processor B at its current frequency. First, show that processor B can be used to its full potential. Indicate which tasks should be assigned to it, give the fixed priorities of the tasks concerned and show that this subset is schedulable with a fixed-priority scheduler.

> We group SENSOR 1 to 3, which makes 100% use ($U_B = 1$). Then unroll the timeline over 20 time units (hyper period) and realize that SENSOR 1 and 2 (higher priority) run for 4 time units every 5 time units, leaving SENSOR 3 (lower priority) time to run for 4 time units every 20 time units.

Question 4. (1 point) Calculate the utilization rate of the tasks still to be executed on processor A. Determine the speed-up factor s that needs to be applied to process A to make them schedulable. For example, a factor $s = 2$ means that the processor runs twice as fast, so budgets are divided by 2. Indicate the priorities of the tasks assigned to it and demonstrate that this subset is schedulable.

> The overall system utilization rate is $U = \frac{7}{3}$. If we remove B's tasks, we are left with $U_A = \frac{4}{3}$. So we need to speed up the processor by $\frac{4}{3}$. The table for the subset of tasks assigned to B becomes:

Task	Period	Processor	Budget
DISPLAY_1	6ms	B	3ms ($4ms * \frac{3}{4}$)
DISPLAY_2	3ms	B	1.5ms ($2ms * \frac{3}{4}$)

> DISPLAY_2 must have higher priority than DISPLAY_1 (smaller period).

3 Multi-processors scheduling (3 points)

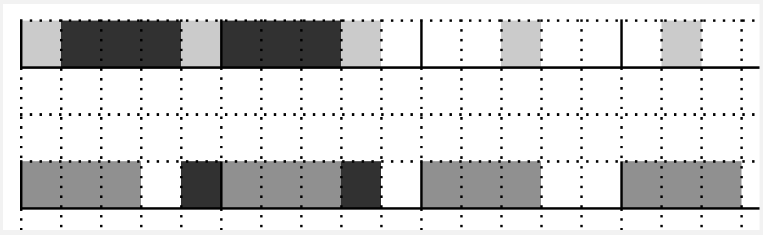
In a system with a two-core processor, tasks are executed according to a Global Deadline Monotonic scheduling policy. Consider the following set of synchronous periodic tasks:

Task	Budget	Deadline	Period
T1	1	4	4
T2	3	5	5
T3	8	9	20

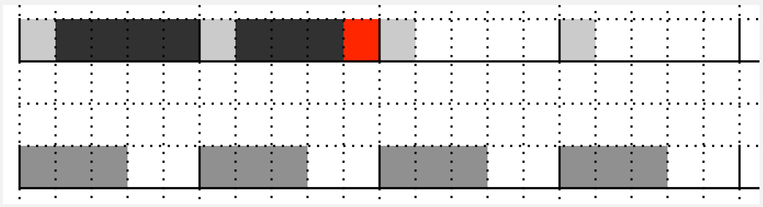
Question 1. (0.5 point) Global Deadline Monotonic scheduling [course]
Recall how Global Deadline Monotonic scheduling works.

course

Question 2. (1 point) Applying Global Deadline Monotonic scheduling to a given task set
Is this task set schedulable ?



Question 3. (1 point) Global Deadline Monotonic scheduling applied to a less loaded task set
T1 period is increased to 5, with the deadline remaining at 4. Recalculate the scheduling over the hyper-period. What do you see now?



Question 4. (0.5 point) Multi-processors scheduling issue [course]
Why are the results so surprising? How would you describe this phenomenon?

course. scheduling anomaly

4 Periodic server for aperiodic tasks (2.5 points)

The following questions only apply to
M2 students from IP Paris master programs
(CPS, CYBER, PDS).

Consider a periodic server of aperiodic tasks with a period of 8ms and a budget of 2ms. This server has the highest priority. An aperiodic task occurs at date 20ms and requires a processing time of 4ms.

Question 1. (1.5 points) Polling server [course]
It is assumed that the server applies a polling server policy. Recall how such a server works. Give the response time of the aperiodic task.

> release at 20, start execution at 24 until 26 and then resume execution at 32 until 34. $R = 34 - 20 = 14$

Assume that the activation date of the sporadic task is different. Give the situation that produces the worst response time.

> release at 1, start execution at 8 until 10 and then resume execution at 16 until 18. $R = 18 - 1 = 17$

Question 2. (1 point) Sporadic server [course]
Assume that the server applies a sporadic server policy. Recall how such a server works. Give the response time of the aperiodic task.

> release at 20, start execution at 20 until 22 and then resume execution at 28 until 30. $R = 30 - 20 = 10$

Part 2 - WCET Analysis

5 Static Program/WCET Analysis (3.5 points)

Question 1. (1.5 points) Explain the three elements that define a data-flow analysis.

- The abstract domain:
  Models the information that should be analyzed.
- The transfer function:
  Models the effect of an instruction on the analysis information.
- The join/meet operator:
  Combines abstract values at control-flow joins.

Question 2. (1 point) In the lecture your have seen a data-flow analysis called *Value Range Analysis*. How can this value range analysis be used during *Worst-Case Execution Time* (WCET) Analysis?

It can be used at different stages of the WCET analysis:

- Loop Bounds:
  For simple, counting, loops VRA can be used to derive loop bounds, i.e., the maximum number of iterations the program may stay inside the loop.
- Memory Addresses:
  VRA also allows to determine the range of addresses accessed by memory load/store instructions, which can be used during cache analysis.

Question 3. (1 point) List the different techniques covered in the lecture that allow to bound the *Worst-Case Execution Time* (WCET) of a program. Provide the advantages/disadvantages of each approach.

- Measurements:
  Observe the actual execution time of running the program on the hardware under realistic conditions. The observations have to cover all real scenarios that may occur during the operation of the real system. The obtained bound may not be safe.
- Probabilistic Analysis:
  Same as above, this time though a probability distribution is fitted to the observations, which under certain conditions allow to obtain safe estimates of the actual probability of a WCET overrun. The conditions (IID) are often not applicable in realistic systems.
- Static Analysis:
  Provides a safe over-approximation of the actual behavior of the program, is amenable to formal proof. The bounds are safe, but might be very pessimistic.

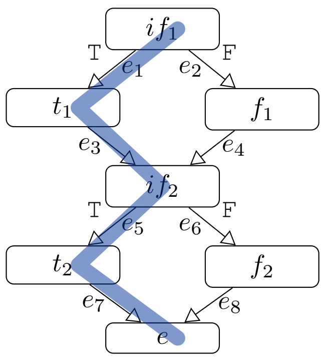
6 Implicit Path Enumeration (4 points)

Question 1. (2 points) Explain what a flow variable is and how it is used by the *Implicit Path Enumeration Technique* (IPET). Give at least two examples on how flow variables are constrained or used in IPET.

A flow variable in IPET expresses the number of times a certain piece of machine code/action is executed by the program. Flow variables are used in IPET to model the flow of execution:

- Kirchhoff's Law:
  The execution flow has to respect Kirchhoff's law, i.e., the number of times a piece of code is entered has to be equal to the number of times the same piece of code is exited.
- Loop Bounds:
  In CFG's with cycles the execution flow could grow infinitely, this has to be prevented using loop bounds. The flow inside the loop is constrained, e.g., by limiting the absolute number of iterations or the relative number of iterations w.r.t. the number of times the loop is entered.
- Cost function:
  Flow variables are also used to model the cost function, i.e., by associating a weight with each flow variable and maximizing the result.

Question 2. (2 points) Consider the *Control-Flow Graph* (CFG) on the right, which consists of two consecutive `if`-statements. The highlighted path indicates that the two `if`-conditions are correlated. More precisely, whenever the condition of the first `if` is `true` the condition of the second `if` will also evaluate to `true`. Provide a *flow fact* for IPET that expresses this correlation. Use the edge names (e_1, e_2, …) as names for the flow variables. Justify your answer.



$$e_1 \leq e_5$$

Whenever the execution enters e_1 it will continue with e_3 and from there has to continue with e_5 – due to the correlation of the two `if`-conditions. Consequently, e_5 has to be at least as large as e_1, or $e_1 \leq e_5$.

7 Cache Analysis (2.5 points)

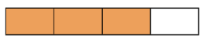
The following questions apply to all students *except* M2 students from IP Paris master programs (CPS, CYBER, PDS).

Question 1. (1.5 points) In the lecture you have seen the `MAY` and `MUST` analyses that are used for *Cache Hit/Miss Classification*. Explain how these two analyses can be used to classify memory accesses into the three categories covered in the lecture based on cache associativity.

The `MUST` analysis provides a maximum age, while the `MAY` analysis provides a minimum age. The classification is then derived depending on the associativity of the cache:

- Always Hit:
  If the maximum age is smaller than the associativity, the access is classified as always hit; the data is guaranteed to be in the cache.
- Always Miss:
  If the minimum age is larger or equal than the associativity, the access is classified as always miss; the data is guaranteed to be not in the cache.
- Not Classified:
  Neither of the two above classifications can be applied; the data may or may not be in the cache.

Question 2. (1 points) The picture below illustrates the analysis information with regard to a *Memory Block* m for *Cache Hit/Miss Classification*. It shows a single *Cache Set* of a 4-way set-associative cache to which m maps. Which of the two illustrations (a) or (b) belongs to the `MAY` analysis? Justify your answer. What is the meaning of the table cells that are highlighted in orange?


(a)

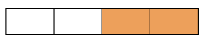

(b)

Illustration (b) concerns `MAY` analysis. The orange cells indicate the possible places in the cache set (when its content is ordered by age) where the memory block m could be placed, i.e., it might have an age of 2, 3, or it might not be in the cache (age of 4). It thus describes the minimum age of the memory block, which is determined by the `MAY` analysis.